

#2

JC997 U.S. PTO
09/942611
08/31/01

520.40578X00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): TANAKA, et al.
Serial No.: Not assigned
Filed: August 31, 2001
Title: VIRTUAL COMPUTER SYSTEM WITH DYNAMIC RESOURCE
REALLOCATION
Group:

LETTER CLAIMING RIGHT OF PRIORITY

Honorable Commissioner of
Patents and Trademarks
Washington, D.C. 20231

August 31, 2001

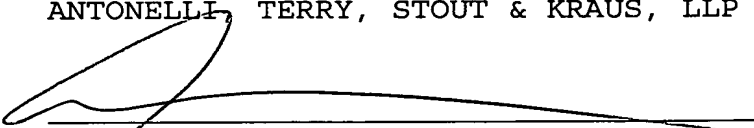
Sir:

Under the provisions of 35 USC 119 and 37 CFR 1.55, the applicant(s) hereby claim(s) the right of priority based on Japanese Patent Application No.(s) 2000-401048 filed December 28, 2000.

A certified copy of said Japanese Application is attached.

Respectfully submitted,

ANTONELLI, TERRY, STOUT & KRAUS, LLP


Alan E. Schiavelli
Registration No. 32,087

AES/amr
Attachment

日 本 国 特 許 庁
JAPAN PATENT OFFICE

JC997 U.S. PTO
09/942611
08/31/01

・ 別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application: 2000年12月28日

出 願 番 号

Application Number: 特願2000-401048

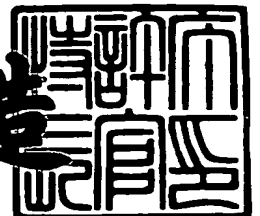
出 願 人

Applicant(s): 株式会社日立製作所

2001年 8月10日

特 許 庁 長 官
Commissioner,
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3072208

【書類名】 特許願

【整理番号】 NT00P0662

【提出日】 平成12年12月28日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/46

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所 中央研究所内

【氏名】 田中 剛

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所 中央研究所内

【氏名】 濱中 直樹

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所 中央研究所内

【氏名】 垂井 俊明

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社日立製作所

【代理人】

【識別番号】 100068504

【弁理士】

【氏名又は名称】 小川 勝男

【電話番号】 03-3661-0071

【選任した代理人】

【識別番号】 100086656

【弁理士】

【氏名又は名称】 田中 恭助

【電話番号】 03-3661-0071

【選任した代理人】

【識別番号】 100094352

【弁理士】

【氏名又は名称】 佐々木 孝

【電話番号】 03-3661-0071

【手数料の表示】

【予納台帳番号】 081423

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 動的な資源分配をする仮想計算機システム

【特許請求の範囲】

【請求項 1】

1 以上の CPU を有する物理計算機上で動作する複数の仮想計算機と、前記複数の仮想計算機のそれぞれにおける前記 CPU の使用率および／または前記複数の仮想計算機のそれぞれにおけるプロセスの待ち行列の長さから前記仮想計算機の負荷状態を監視する負荷状態監視部と、前記複数の仮想計算機への物理資源の割り当てを動的に変更する再構成部と、前記負荷状態監視部によって得られた負荷状態に基づき前記仮想計算機への物理資源の割り当てを求め前記再構成部に再構成を要求する制御部とを備えたことを特徴とする仮想計算機システム。

【請求項 2】

1 以上の CPU を有する物理計算機上で動作し前記 CPU の使用率および／又はプロセスの待ち行列の長さを測定する OS をそれぞれ有する複数の仮想計算機と、前記仮想計算機を制御し前記 OS から測定された情報から前記複数の仮想計算機の負荷状態を監視し監視状態に従って前記複数の仮想計算機への物理資源の割り当てを動的に行なうハイパーバイザとを備えたことを特徴とする仮想計算機システム。

【請求項 3】

1 以上の CPU を有する物理計算機上で動作し前記 CPU の使用率および／又はプロセスの待ち行列の長さを測定する OS をそれぞれ有する複数の仮想計算機と、前記複数の仮想計算機を制御し前記複数の仮想計算機への物理資源の割り当てを動的に変更する再構成部を含むハイパーバイザと、前記複数の仮想計算機のうちの第 1 の仮想計算機で動作し前記複数の仮想計算機のうちの第 2 の仮想計算機上で動作する OS で測定された情報を取得しその情報に基づいて前記再構成部に対して再構成を要求する監視部とを備えたことを特徴とする仮想計算機システム。

【請求項 4】

1 以上の CPU を有する複数の物理計算機と、前記複数の物理計算機の第 1 の

物理計算機上で構成され当該物理計算機上で動作し前記CPUの使用率および／又はプロセスの待ち行列の長さを測定するOSをそれぞれ有する複数の仮想計算機と、第1の物理計算機上で動作し前記複数の仮想計算機を制御し前記複数の仮想計算機への物理資源の割り当てを動的に変更する再構成部を含むハイパーバイザと、前記複数の物理計算機のうちの第2の物理計算機で動作し前記第1の物理計算機上で動作するOSで測定された情報を取得しその情報に基づいて前記再構成部に対して再構成を要求する監視部とを備えたことを特徴とする仮想計算機システム。

【請求項5】

前記OSは稼動状態でCPUの増減が可能であり、前記ハイパーバイザは前記測定された情報によって稼動CPUの数を増または減の処理を行なうことを特徴とする請求項2または3または4のいずれか記載の仮想計算機システム。

【請求項6】

1以上のCPUを有する物理計算機上で動作する複数の仮想計算機と、前記複数の仮想計算機のそれぞれにおける前記CPUの動作状況から前記仮想計算機の負荷状態を監視する負荷状態監視部と、前記複数の仮想計算機への物理資源の割り当てを動的に変更する再構成部と、前記負荷状態監視部によって得られた負荷状態に基づき前記仮想計算機への物理資源の割り当てを求め前記再構成部に稼動するCPUの数の増減を含む再構成を要求する制御部とを備えたことを特徴とする仮想計算機システム。

【請求項7】

前記制御部は負荷の高い仮想計算機に対して、他の仮想計算機のCPU使用率の小ささによって、前記他の仮想計算機が前記負荷の高い仮想計算機に引き渡すCPU割り当て時間の割合を大きくするよう再構成方針を生成することを特徴とする請求項1又は6記載の仮想計算機システム。

【請求項8】

1以上のCPUおよび主記憶装置を有する物理計算機上で動作する複数の仮想計算機と、前記複数の仮想計算機のそれぞれにおける前記主記憶装置の負荷状況から前記仮想計算機の負荷状態を監視する負荷状態監視部と、前記複数の仮想計

算機への物理資源の割り当てを動的に変更する再構成部と、前記負荷状態監視部によって得られた負荷状態に基づき前記仮想計算機への物理資源の割り当てを求め前記再構成部に再構成を要求する制御部とを備えたことを特徴とする仮想計算機システム。

【請求項 9】

前記主記憶装置の負荷状況はページングおよび／又はスワップの頻度によって求められ、前記再構成部は前記主記憶装置の前記仮想計算機への領域の割当量を動的に変更することを特徴とする請求項 8 記載の仮想計算機システム。

【請求項 10】

1 以上の CPU および主記憶装置を有する物理計算機上で動作し前記主記憶装置の負荷状態を測定する OS をそれぞれ有する複数の仮想計算機と、前記仮想計算機を制御し前記 OS から測定された情報から前記複数の仮想計算機の負荷状態を監視し監視状態に従って前記複数の仮想計算機への物理資源の割り当てを動的に行なうハイパーバイザとを備えたことを特徴とする仮想計算機システム。

【請求項 11】

1 以上の CPU および主記憶装置を有する物理計算機上で動作し前記主記憶装置の負荷状況を測定する OS をそれぞれ有する複数の仮想計算機と、前記複数の仮想計算機を制御し前記複数の仮想計算機への物理資源の割り当てを動的に変更する再構成部を含むハイパーバイザと、前記複数の仮想計算機のうちの第 1 の仮想計算機で動作し前記複数の仮想計算機のうちの第 2 の仮想計算機上で動作する OS で測定された情報を取得しその情報に基づいて前記再構成部に対して再構成を要求する監視部とを備えたことを特徴とする仮想計算機システム。

【請求項 12】

1 以上の CPU および主記憶装置を有する複数の物理計算機と、前記複数の物理計算機の第 1 の物理計算機上で構成され当該物理計算機上で動作し前記主記憶装置の負荷状況を測定する OS をそれぞれ有する複数の仮想計算機と、第 1 の物理計算機上で動作し前記複数の仮想計算機を制御し前記複数の仮想計算機への物理資源の割り当てを動的に変更する再構成部を含むハイパーバイザと、前記複数の物理計算機のうちの第 2 の物理計算機で動作し前記第 1 の物理計算機上で動作

するOSで測定された情報を取得しその情報に基づいて前記再構成部に対して再構成を要求する監視部とを備えたことを特徴とする仮想計算機システム。

【請求項 1 3】

1 以上のCPUを有する物理計算機上で動作しそれぞれアプリケーションプログラムの実行を制御するOSを有する複数の仮想計算機と、前記複数の仮想計算機のそれぞれにおける前記アプリケーションプログラムの処理の応答時間から前記仮想計算機の負荷状態を監視する負荷状態監視部と、前記複数の仮想計算機への物理資源の割り当てを動的に変更する再構成部と、前記負荷状態監視部によって得られた負荷状態に基づき前記仮想計算機への物理資源の割り当てを求め前記再構成部に再構成を要求する制御部とを備えたことを特徴とする仮想計算機システム。

【請求項 1 4】

1 以上のCPUを有する物理計算機上で動作しそれぞれアプリケーションプログラムの実行を制御するOSを有する複数の仮想計算機と、前記仮想計算機を制御し前記アプリケーションプログラムから得られた処理の応答時間から前記複数の仮想計算機の負荷状態を監視し監視状態に従って前記複数の仮想計算機への物理資源の割り当てを動的に行なうハイパーバイザとを備えたことを特徴とする仮想計算機システム。

【請求項 1 5】

1 以上のCPUを有する物理計算機上で動作しそれぞれアプリケーションプログラムの実行を制御するOSを有する複数の仮想計算機と、前記複数の仮想計算機を制御し前記複数の仮想計算機への物理資源の割り当てを動的に変更する再構成部を含むハイパーバイザと、前記複数の仮想計算機のうちの第1の仮想計算機で動作し前記複数の仮想計算機のうちの第2の仮想計算機上で動作する前記アプリケーションプログラムから得られた処理の応答時間を取得しその情報に基づいて前記再構成部に対して再構成を要求する監視部とを備えたことを特徴とする仮想計算機システム。

【請求項 1 6】

1 以上のCPUを有する複数の物理計算機と、前記複数の物理計算機の第1の

物理計算機上で構成され当該物理計算機上で動作しそれぞれアプリケーションプログラムの実行を制御するOSを有する複数の仮想計算機と、第1の物理計算機上で動作し前記複数の仮想計算機を制御し前記複数の仮想計算機への物理資源の割り当てを動的に変更する再構成部を含むハイパーバイザと、前記複数の物理計算機のうちの第2の物理計算機で動作し前記第1の物理計算機上で動作するアプリケーションプログラムの処理の応答時間を取得しその情報に基づいて前記再構成部に対して再構成を要求する監視部とを備えたことを特徴とする仮想計算機システム。

【請求項17】

前記負荷状態監視部はアプリケーションプログラムに対して、トランザクションを発行してそのトランザクションが完了するまでの時間に基づいて前記仮想計算機の負荷状態を監視することを特徴とする請求項13記載の仮想計算機システム。

【請求項18】

1以上のCPUを有する物理計算機上で動作しそれぞれアプリケーションプログラムの実行を制御するOSを有する複数の仮想計算機と、前記仮想計算機を制御し前記アプリケーションプログラムにトランザクションを発行しそのトランザクションが完了するまでの時間を監視することにより得られた処理の応答時間から前記複数の仮想計算機の負荷状態を監視し監視状態に従って前記複数の仮想計算機への物理資源の割り当てを動的に行なうハイパーバイザとを備えたことを特徴とする仮想計算機システム。

【請求項19】

1以上のCPUを有する物理計算機上で動作しそれぞれアプリケーションプログラムの実行を制御するOSを有する複数の仮想計算機と、前記複数の仮想計算機を制御し前記複数の仮想計算機への物理資源の割り当てを動的に変更する再構成部を含むハイパーバイザと、前記複数の仮想計算機のうちの第1の仮想計算機で動作し前記複数の仮想計算機のうちの第2の仮想計算機上で動作する前記アプリケーションプログラムに対してトランザクションを発行しそのトランザクションが完了するまでの時間を監視することから得られた処理の応答時間を取得しそ

の情報に基づいて前記再構成部に対して再構成を要求する監視部とを備えたことを特徴とする仮想計算機システム。

【請求項 2 0】

1 以上の CPU を有する複数の物理計算機と、前記複数の物理計算機の第 1 の物理計算機上で構成され当該物理計算機上で動作しそれぞれアプリケーションプログラムの実行を制御する OS を有する複数の仮想計算機と、第 1 の物理計算機上で動作し前記複数の仮想計算機を制御し前記複数の仮想計算機への物理資源の割り当てを動的に変更する再構成部を含むハイパーバイザと、前記複数の物理計算機のうちの第 2 の物理計算機で動作し前記第 1 の物理計算機上で動作するアプリケーションプログラムに対してトランザクションを発行しそのトランザクションの完了するまでの時間を取得しその情報に基づいて前記再構成部に対して再構成を要求する監視部とを備えたことを特徴とする仮想計算機システム。

【請求項 2 1】

1 以上の CPU を持つ物理計算機上で動作する複数の仮想計算機と、前記複数の仮想計算機の負荷状態を監視する負荷状態監視部と、前記負荷状態監視部によって負荷が高いと判断された仮想計算機に配分された物理資源を変更する複数の対策内容を記憶する記憶部と、前記複数の対策内容を順次実行し負荷低減の効果のある対策内容で物理資源の再構成を行なう手段とを備えたことを特徴とする仮想計算機システム。

【請求項 2 2】

物理計算機上で動作する複数の仮想計算機と、前記複数の仮想計算機に前記物理計算機の物理資源を動的に割り当てる再構成部と、少なくとも 1 つの前記仮想計算機の負荷を定間隔で採取し採取した負荷データの周期的変化を検出する負荷状態監視部と、前記負荷の周期的変化に基づき前記物理資源の割り当てを決定し前記再構成部に周期的に物理資源の割り当ての再構成を要求する制御部とを備えたことを特徴とする仮想計算機システム。

【請求項 2 3】

物理計算機上で動作する複数の仮想計算機と、前記複数の仮想計算機に前記物理計算機の物理資源を動的に割り当てる再構成部と、顧客との契約条件に従って

前記再構成部での物理資源の各仮想計算機への割り当ての優先順位を決める制御部とを備えたことを特徴とする仮想計算機システム。

【請求項 2 4】

前記制御部は各仮想計算機毎に顧客との契約条件に従って異なる過負荷と判断する基準を有していることを特徴とする請求項 2 3 記載の仮想計算機システム。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、仮想計算機システムに関し、特に仮想計算機への資源の割り当てを仮想計算機の負荷に応じて動的に再構成する技術に関するものである。

【0 0 0 2】

【従来の技術】

仮想計算機システムにおいては、CPU、主記憶、IO等の物理資源が論理的に区画化され、仮想計算機システム上に実現する各仮想計算機LPAR(Logical Partition)に割り当てられる。仮想計算機システムにおいて、物理資源を動的に仮想計算機に割り当てる機構に関しては、特開平6-103092号公報、特開平6-110715号公報、で記述されている。これらの文献に示されている仮想計算機システムでは、LPARの物理資源の配分を変更をする場合、オペレータによる操作、あるいは時間駆動事象(time driven: タイマがある時刻になったことで駆動する)で再構成要求が、仮想計算機システム全体を制御するプログラム(ハイパーバイザ)に対して発行される。ハイパーバイザは、オペレータが再構成要求発行前に設定しておいた資源配分方法に従って、LPARの構成を動的に変更する。

【0 0 0 3】

また、ハイパーバイザには、各LPARのCPU使用時間等のシステム動作状況を収集するモニタ機能もある。これらの装置では、オペレータが物理資源の配分を決定する必要があり、システムの動作状況から自動的に資源を配分した後再構成を自動的に行うことはできなかった。ところが、特開平9-26889号公報の記載では、あるLPARが同一仮想計算機システム内の別のLPARのCP

U使用時間をハイパーバイザに問い合わせ、実際のCPU使用時間と設定されたCPU使用時間に差がある場合はCPU割り当て時間を実際のCPU使用時間に合わせるようにする装置を提案している。しかしながら、実際の計算機システムの負荷を考えた場合CPUの使用時間で計算機システムの負荷を知ることは困難である。また、CPU使用時間を増大させるだけでシステムの応答性能を改善することは困難である。

【0004】

【発明が解決しようとする課題】

このような単純なケースではなく、Webサーバやアプリケーションサーバといったアプリケーションでの応答時間といったCPU使用時間以外の計算機の負荷と、その負荷にあわせた計算機の物理資源を自動的に調整する方法は提案されていない。自動的に資源分配を行うことの優れた点は例えば以下の事例である。データセンター（インターネット・ビジネス用のサーバを設置し、その運用を引き受けるサービス）といった用途で計算機を使用する場合、管理しなければならない計算機の台数は膨大な数となる。このとき、仮想計算機システムの物理資源を有効に使用するために各LPARの負荷に適応して物理資源の配分を増減することが自動的に行なわれれば管理のコスト低減や、システムの性能保証といった面で有効であると考えられる。

【0005】

そこで、本発明は仮想計算機システムのアプリケーションやOSで観測されるLPARの負荷状況に適合したLPARの再構成を実行する仮想計算機システムを提供することを目的とする。

【0006】

【課題を解決するための手段】

前記目的達成のために本発明は、物理計算機上で複数のLPARが動作し、ハイパーバイザによって各LPAR間でCPUや主記憶といった物理資源の割り当てを動的に再構成し、各LPARのCPU使用率又はプロセスの実行待ち行列長や主記憶のスワップ頻度といったシステムの負荷や、アプリケーションプログラムの処理応答時間を測定する手段を有する仮想計算機システムであって、該負荷

測定手段によって測定された L P A R の負荷に基づき、各 L P A R ごとに割り当てている物理資源の量を変更し、L P A R の再構成を行う。

【 0 0 0 7 】

【発明の実施の形態】

本発明の実施例においては次のような処理が行なわれる。

物理計算機上で複数の L P A R が動作し、ハイパーバイザによって各 L P A R 間で C P U や主記憶といった物理資源の割り当てを動的に再構成し、各 L P A R の C P U 使用率、プロセスの実行待ち行列長や主記憶のスワップ頻度といったシステムの負荷やアプリケーションプログラムの処理応答時間を測定する手段を備え、この負荷測定手段によって測定された L P A R の負荷に基づき、各 L P A R ごとに割り当てている物理資源の量を変更し、L P A R の再構成を行う。

【 0 0 0 8 】

さらに、L P A R 上で動作する O S として、稼動時に動的に C P U 数を変更したり主記憶量を変更できる機能を有する O S を用い、C P U の数や主記憶量を L P A R の負荷に応じた L P A R の再構成を行う。

【 0 0 0 9 】

また、より効果的な物理資源配分を実現するために再構成後の各 L P A R の負荷を計測し、再構成前に高負荷であった L P A R の負荷が低下したかどうか判定し、効果が無かった場合は再構成前の構成に戻すことで適切な L P A R の再構成を行う。

【 0 0 1 0 】

同様に、効果的な物理資源配分のために、仮想計算機の負荷の変化を監視する。そこで、周期的な負荷の変化が見られる場合は、高負荷時には C P U の割り当て時間や C P U 数等の物理資源を増強し、低負荷時には他の高負荷 L P A R に物理資源を配分し、負荷状況に合わせて構成を周期的に変更する。

【 0 0 1 1 】

以下、本発明に係わる仮想計算機システムの実施例を図面を用いて説明する。

図 2 に、本発明の全実施例で共通な 1 つの仮想計算機システムを構成するための 1 つの物理計算機システム構成を示す。この物理計算機システムが複数個あ

て良い。図2が本発明でいうところの1つの仮想計算機システムを構成する物理計算機である1つの密結合マルチプロセッサを示す。10、11、・・・、1nは、それぞれ物理プロセッサCPU0、CPU1、・・・、CPUnを示している。20は主記憶装置を示している。30、31、・・・、3mはそれぞれ、I/O装置I/O0、I/O1、・・・、I/Omを示している。40は、主記憶に常駐して仮想計算機システム全体を制御するハイパーバイザを示している。

【0012】

図3に仮想計算機システムの概要を示す。これは図2に示す物理計算機システムに対応する1つの仮想計算機システムを示す。40は、ハイパーバイザを示している。50、・・・、5kは仮想計算機LPAR0、・・・、LPARkを示している。50-0、・・・、50-nはLPAR0に含まれる論理プロセッサLP0、・・・、LPnを示し、5k-0、・・・、5k-nはLPARkに含まれる論理プロセッサLP0、・・・、LPnを示している。各LPARが複数の論理プロセッサLPを含むのは物理構成がマルチプロセッサシステムだからである。

【0013】

図4に、主記憶装置20の概要を示す。主記憶装置20には、ハイパーバイザおよび各LPARのそれぞれに割り当てられている領域がある。

【0014】

図5にLPAR情報テーブル100を示す。LPAR情報テーブル100は、各LPARの物理資源の割り当てを示している。101はLPAR名を表す欄、102は各LPARに割り当てられた主記憶装置上の領域の開始アドレスを示す。103は各LPARの物理主記憶容量を定義する欄である。104は各LPARに割り当てているCPUの割り当て比率を定義する欄を示している。CPU割り当て比率に基づき、ハイパーバイザ40は、CPU10、・・・、1nを、LPAR0、・・・、LPARkに時分割で割り当てる。

【0015】

図6にハイパーバイザの構成を示す。ハイパーバイザ40は、各LPARのスケジューリングを行うスケジューラ200、各LPARに分配される物理資源を

管理する資源管理部201、各LPARへの操作コマンド等を制御するLPAR制御部202、各LPARのオペレーティングシステムが実行される論理プロセッサ204を制御する論理プロセッサ制御部203、オペレータがLPARを操作する情報を入力するためのシステムコンソールの画面であるフレームや、LPARの状態をオペレータに通知する情報を持つ画面であるフレームを制御するフレーム制御部205、LPARの物理資源割り当てを計画する再構成方針制御部206、各LPARにかかっている負荷状況を監視する負荷状態監視部207から構成される。

【0016】

以下、本実施例に係わる仮想計算機システムの動作を、CPUを4個備えた物理計算機上で3台のLPARを稼働させる場合を例にとり説明する。ここでは、図7に示すような割り当てでCPUを時分割、あるいは排他的に使用していると仮定する。即ち、CPU0は100%LPAR0に使用され、CPU1はLPAR0とLPAR1に50%ずつ時分割で使用され、CPU2、CPU3はそれぞれLPAR1とLPAR2により100%ずつ使用される。

【0017】

なお、本発明での再構成方針制御部206、負荷状態監視部207は、ハイパーバイザの一部として実装されることに限定されず、OS上で動作するユーザプログラムとして実装されてもよい。再構成方針制御部206、負荷状態監視部207の機能をもつプログラムが動作する計算機を以降、ポリシーサーバと呼ぶことにする。ポリシーサーバは、図15に示すように負荷状況の調査対象のLPAR50、・・・、5kが動作している仮想計算機システム上の特定のLPAR5xであってもよい。また、図16の様にネットワーク61で接続した物理計算機60-1と物理計算機60-xにおいて物理計算機60-1上で動作しているLPARの負荷を計測するポリシーサーバを物理計算機60-xに実装してもよい。物理計算機60-xでは、単一OSが動作していても、複数のLPARが動作していてもよい。LPAR5kや物理計算機60-xはポリシーサーバの専用ではなく、他のアプリケーションの処理を行なうことが出来る。

以上で、本発明の実施例の前提となるシステム構成の説明を終わり、各具体的

な実施例の説明をする。

【 0 0 1 8 】

(実施例 1)

以降、図 1 を用いて各 L P A R 上の O S が計測する L P A R の負荷状態を調査し動的再構成を行なうまでの流れを説明する。ここで、負荷状態とは C P U の使用率やプロセスの実行待ち行列長を表わす。

【 0 0 1 9 】

仮想計算機システムの操作者(以降、オペレータ)はフレームで L P A R の負荷状態を調査する要求と、負荷を調査する時間間隔を設定する。フレーム制御部 2 0 5 は、スケジューラ 2 0 0 を通し負荷状態監視部 2 0 7 へ L P A R 負荷状態の監視要求と監視間隔を通知する(3 0 0、3 0 1)。そして、負荷状態監視部 2 0 7 は、スケジューラ 2 0 0 を通して L P A R 制御部 2 0 2 に負荷状態調査要求(3 0 2、3 0 3)を通知する。L P A R 制御部 2 0 2 は、各論理プロセッサ制御部 2 0 3 に対して各論理プロセッサ 2 0 4 の負荷状態を調査し(3 0 5)、調査結果(3 0 6、3 0 7)を負荷状態監視部 2 0 7 に転送する要求 3 0 4 を発行する。負荷状態監視部 2 0 7 は、各 L P A R の負荷状態を負荷状態監視部 2 0 7 内部で保存する。負荷状態の情報の保存量は、オペレータがフレームを通してフレーム制御部 2 0 5 に指示し、スケジューラ 2 0 0 を通して負荷状態監視部 2 0 7 へ通知される。

【 0 0 2 0 】

本実施例では、負荷状態を表す数値として C P U 使用率と実行待ちのプロセス数を示す実行待ち行列長を使用する。図 8 に各 L P A R の負荷状態の例を示す。これは L P A R 毎に各 C P U の使用率とタスクまたはスレッドの実行待ち行列長を示している。これらの情報を採取する要求 3 1 0 は、L P A R 制御部 2 0 2 から各 L P A R の論理プロセッサ制御部 2 0 3 に通知される。論理プロセッサ制御部 2 0 3 は、論理プロセッサ 2 0 4 を通し各 L P A R 上で稼動している O S 1 へ割り込みを行い、O S 1 の動作状況に関するカウンタから、C P U 使用率や実行待ち行列長の情報の取得を要求し(3 1 3)、負荷状態情報を取得する(3 1 1、3 1 2)。論理プロセッサ制御部 2 0 3 は調査結果(3 0 6、3 0 7)を負荷状態

監視部 207 に転送する。

【0021】

一般に CPU 利用率が高く、実行待ち行列長が長ければシステムの負荷状態が高いと考えられる。そこで、負荷状態監視部 207 では一定期間（例えば 1 時間）の負荷状態の平均を算出し、オペレータがフレームで設定した閾値を超えた場合、再構成方針生成部 206 に対して各 LPAR の再構成要求を発行する（320）。

【0022】

再構成方針生成部 206 は、各 LPAR の負荷状態を負荷状態監視部 207 から読み出し（330）、現在の各 LPAR への CPU 割り当てを資源管理部 201 から読み出す（331）。次に、再構成方針生成部 206 は、負荷状態と現在の CPU 割り当てから再構成方針テーブル 900（図 9）を内部に生成する。これは図 7 に示し資源管理部 201 に格納されている現在の CPU の割り当て状況からみて、LPAR0 が CPU の負荷が高いと判断し、LPAR0 への CPU 割り当てを増やした状態を示している。

【0023】

再構成の方針は、システムの負荷情報の中で注目する物によって異なってくる。CPU がネックであれば、CPU の割り当て時間を増やしたり CPU 数を増やす対策が考えられる。また、各 LPAR 上で動作する OS が稼動時に CPU 数を増減できる機能を持っているかどうかで対策が異なる。OS には OS が稼動中に新たな CPU を起動させることが出来るものと、OS の動作を一旦リセットし、その後再構成をして、起動する CPU の数を変更しなければならないものがある。OS が稼動時に CPU 数の増減ができない場合は、CPU の割り当て時間を変化させるのみである。しかし、OS 稼動時に CPU 数が変更できる場合は、CPU の割り当て時間と CPU 数を変更することが可能となる。本実施例では稼動時 CPU 数再構成可能の機能がある OS を使用する。また、OS 稼動時に主記憶の容量を変更することが可能な OS を使用している場合、負荷状態として主記憶のページング（実メモリ上でのページの書き替え）やスワップ（アプリケーションプログラムの入れ替え）の頻度を調査し、頻度が高い場合は主記憶量を増やす対

策をすることができる。本実施例ではCPUの負荷状態に基づいて各LPARの再構成を行なう例を示す。

【 0 0 2 4 】

再構成方針生成部 2 0 6 の内部には、注目する負荷の種類、負荷が重いと判断する閾値、対策する負荷の優先度、負荷が高い場合の対策の対応表(図 1 0)がある。この表は、オペレータがフレームで設定し、フレーム制御部 2 0 5 からスケジューラ 2 0 0 経由で再構成方針生成部 2 0 6 に書き込みがあった通知をする(3 4 0、3 4 1)。この通知を受け取った再構成方針生成部 2 0 6 はフレーム制御部 2 0 5 内のデータを読み出し(3 4 2、3 4 3)、再構成方針生成部 2 0 6 内部の対応表に書き込む。

【 0 0 2 5 】

この実施例では、図 1 0 の対策表に基づき、LPAR 0 のCPU使用率が高いためCPUの割り当て時間を増やし、実行待ち行列長が長いのでLPAR 0 に割り当てるCPU数を増加させ同時実行可能なプロセス数を増やすことで負荷を下げる対策をとる。このような物理資源の移動で負荷の低いLPARの性能を落としてしまっては意味が無い為、一例として図 1 1 のようなCPUの平均使用率ごとに、負荷の軽いLPARのCPUから何%のCPU時間を他の負荷が重いLPARに割り当てる方法を適用する。図 1 1 にはLPARの現在のCPU使用率が小さいほど他のLPARに割り当てる率を大きくしたものが記載されている。ここでは、現在のCPU利用率が低いLPARからでもそのCPUの割り当ての全部またはほとんどを他のLPARに割り当てることにより逆にそのLPARの負荷が高くなってしまふことを防いでいる。この割り当てに基づいて再構成の方針を作成したのが図 9 の再構成方針テーブル 9 0 0 である。

【 0 0 2 6 】

再構成方針生成部 2 0 6 は、スケジューラ 2 0 0 に対して再構成要求を発行する(3 5 0)。同時に、負荷状態監視部 2 0 7 に対して性能計測の停止要求を発行する(3 5 1)。

【 0 0 2 7 】

LPARの再構成手順については、従来では再構成は、オペレータによる操作

、あるいは時間駆動事象の契機で行われるが、本発明では、ハイパーバイザがシステムの負荷が閾値を超えたというイベントで再構成要求を発行して行なわれる。

【 0 0 2 8 】

まず、スケジューラ 2 0 0 は、再構成方針生成部 2 0 6 内部の図 9 の再構成方針テーブル 9 0 0 を読み出し (3 8 0) 、資源管理部 2 0 1 内部の L P A R 情報テーブルを書き換え (3 8 1) 、各 L P A R 制御部 2 0 2 へ構成変更を指示する。

【 0 0 2 9 】

L P A R 制御部 2 0 2 は、再構成する L P A R に属する論理プロセッサ 2 0 4 の O S 1 をストップ状態にさせる (3 6 0 、 3 6 1 、 3 6 2) 。次に L P A R 制御部 2 0 2 が資源管理部 2 0 1 の L P A R 情報テーブル読み出し要求を発行し (3 6 4) 、読み出した構成 (3 6 5) を、 L P A R 制御部 2 0 2 の内部に保存する。

【 0 0 3 0 】

L P A R 制御部 2 0 2 は、論理プロセッサ 2 0 4 へ O S の再稼動を各論理プロセッサ制御部 2 0 3 へ指示する (3 7 0 、 3 7 1 、 3 7 2) 。 O S 1 の再起動後、 L P A R 制御部は、 L P A R 再構成終了を再構成方針生成部 2 0 6 負荷状態監視部 2 0 7 へ通知する (3 7 5) 。負荷状態監視部 2 0 7 は L P A R の負荷状態の調査要求を前記の通り L P A R 制御部 2 0 2 へ発行する (3 0 2 、 3 0 3) 。以上の処理で資源の割り当て変更は終了する。

【 0 0 3 1 】

時分割された C P U の割り当て時間を変更した場合、スケジューラ 2 0 0 が新しく定義された C P U 割り当て時間を実行すればよく、構成変更はハイパーバイザ内の処理のみで完了する。新たに C P U (論理プロセッサ) を追加する場合、 L P A R 制御部 2 0 2 が、論理プロセッサ制御部 2 0 3 を介して、もしくは直接に、新たに割り当てられた C P U (論理プロセッサ) を割り込み等で L P A R 上の O S に通知し、 L P A R 上の O S が自発的に新しく追加された C P U (論理プロセッサ) を起動するコマンドを、対応する L P A R 制御部 2 0 2 に送る。

【 0 0 3 2 】

以上のように、L P A R の O S から C P U の稼働率および／またはプロセスの待ち行列の長さを取得することにより、C P U の割り当て時間や主記憶の容量などの資源の割り当てを変更する。これにより、C P U の使用時間を計測するより正確に L P A R の負荷の大きさが把握できる。更に、オペレータの逐次の指示なしで負荷の高い L P A R に資源を多く割り当てることが出来る。

【0033】

(実施例2)

以降、図17を用いて L P A R 上で動作するアプリケーションの負荷状態を調査し動的再構成を行なうまでの流れを説明する。ここで、アプリケーションの負荷状態とは、アプリケーションプログラムの処理の応答時間を指す。例えば、データベースから表を取り出し、表の内容を更新するといったトランザクション処理の応答時間のことを指している。

【0034】

仮想計算機システムのオペレータはフレームで L P A R の負荷状態を調査する要求と、負荷を調査する時間間隔を設定する。フレーム制御部205は、スケジューラ200を通し負荷状態監視部207へ L P A R 負荷状態の監視要求と監視間隔を通知する(300、301)。そして、負荷状態監視部207は、スケジューラ200を通して L P A R 制御部202に負荷状態調査要求(302、303)を通知する。L P A R 制御部202は、各論理プロセッサ制御部203に対して各論理プロセッサ204の負荷状態を通知された監視間隔で調査し(305)、調査結果(306、307)を負荷状態監視部207に転送する要求304を発行する。負荷状態監視部207は、各 L P A R の負荷状態を負荷状態監視部207内部で保存する。負荷状態の情報の保存量は、オペレータがフレームを通してフレーム制御部205に指示し、スケジューラ200を通して負荷状態監視部207へ通知される。

【0035】

アプリケーションの負荷状態を採取する要求310は、L P A R 制御部202から各 L P A R の論理プロセッサ制御部203に通知される。論理プロセッサ制御部203は、論理プロセッサ204、O S 1 に割り込み信号を出し、O S 1 か

らアプリケーション400の負荷状態の情報をアプリケーション400に要求するシグナルをアプリケーション400に送る(313、314)。アプリケーション400からの負荷状態は論理プロセッサ制御部203を通し(315、311、312)、負荷状態監視部207に転送される(307)。また、アプリケーションの応答時間だけでなく同時に実施例1で示したCPUの負荷状態も負荷状態監視部207に転送される。

【0036】

負荷状態監視部207では一定期間(例えば1時間)のアプリケーションの応答時間の平均を算出する(図18)。アプリケーションプログラムにトランザクションを受け取ってから完了するまでの時間の計測手段を持たせることによって応答時間を計測する。応答時間がオペレータがフレームで設定しておいた閾値を超えたことを契機に、再構成方針生成部206に対して各LPARの再構成要求を発行する(320)。例えば、応答時間の閾値を5秒とした場合、図18の応答時間分布ではLPAR0の物理資源を増加させ性能の改善を図るように再構成する。

【0037】

再構成方針生成部206は、各LPARの負荷状態を負荷状態監視部207から読み出し(330)、現在の各LPARへのCPU割り当てを資源管理部201から読み出す(331)。次に、再構成方針生成部206は、負荷状態と現在のCPU割り当てから再構成方針テーブル900(図9)を内部に生成する。再構成方針生成部206の内部には、注目する負荷の種類、負荷が重いと判断する閾値、対策する負荷の優先度、負荷が高い場合の対策の対応表(図10)がある。この表は、オペレータがフレームで設定し、フレーム制御部205からスケジューラ200経由で再構成方針生成部206に書き込みがあった通知をする(340、341)。この通知を受け取った再構成方針生成部206はフレーム制御部205内のデータを読み出し(342、343)、再構成方針生成部206内部の対応表に書き込む。

【0038】

再構成の方針は、システムの負荷状態の中で注目する情報によって異なってくる

る。本実施例では、CPUの使用時間の分布からLPAR0に追加するCPUの割り当て時間を算出する実施例1と同様の処理を行う。

【0039】

まず、図10の対策表に基づき、LPAR0のCPU使用率が高いためCPUの割り当て時間を増やし、実行待ち行列長が長いのでLPAR0に割り当てるCPU数を増加させ同時実行可能なプロセス数を増やすことで負荷を下げる対策をとる。このような物理資源の移動で負荷の低いLPARの性能を落としてしまっは意味が無い為、一例として図11のようなCPUの平均使用率ごとに、負荷の軽いLPARのCPUから何%のCPU時間を他の負荷が重いLPARに割り当てる方法を適用する。この割り当てに基づいて再構成の方針を作成したのが図9の表である。

【0040】

再構成方針生成部206は、スケジューラ200に対して再構成要求を発行する(350)。同時に、負荷状態監視部207に対して性能計測の停止要求を発行する(351)。

【0041】

LPARの再構成手順において、再構成は、従来、オペレータによる操作、あるいは時間駆動事象の契機で行なわれるが、本発明では、ハイパーバイザがシステムの負荷が閾値を超えたというイベントで再構成要求を発行する。

【0042】

まず、スケジューラ200は、再構成方針生成部206内部の再構成方針テーブルを読み出し、資源管理部201内部のLPAR情報テーブルを書き換え、各LPAR制御部202へ構成変更を指示する。

【0043】

LPAR制御部202は、再構成するLPARに属する論理プロセッサ204のOS1をストップ状態にさせる(360、361、362)。次にLPAR制御部202が資源管理部201のLPAR情報テーブル読み出し要求を発行し(364)、読み出した構成(365)を、LPAR制御部202の内部に保存する。

【0044】

L P A R 制御部 202 は、論理プロセッサ 204 へ O S の再稼動を各論理プロセッサ制御部 203 へ指示する(370、371、372)。O S 1 の再起動後、L P A R 制御部は、L P A R 再構成終了を再構成方針生成部、206 負荷状態監視部 207 へ通知する(375)。負荷状態監視部 207 は L P A R の負荷状態の調査要求を前記の通り L P A R 制御部 202 へ発行する(302、303)。以上の処理で資源の割り当て変更は終了する。

【0045】

時分割された C P U の割り当て時間を変更した場合、スケジューラ 200 が新しく定義された C P U 割り当て時間を実行すればよく、構成変更はハイパーバイザ内の処理のみで完了する。新たに C P U (論理プロセッサ)を追加する場合、L P A R 制御部 202 が、論理プロセッサ制御部 203 を介して、もしくは直接に、新たに割り当てられた C P U (論理プロセッサ)を割り込み等で L P A R 上の O S に通知し、L P A R 上の O S が自発的に新しく追加された C P U (論理プロセッサ)を起動するコマンドを、対応する L P A R 制御部 202 に送る。以上で再構成が終了する。そして、負荷状態監視部 207 は再び各 L P A R の負荷状態の監視を再開する。

【0046】

以上のように、アプリケーションプログラムでの応答時間の長さから C P U の負荷の大きさを判断し、資源の割り当てを行なうことにより、より正確に負荷が大きいかどうかの稼動状態が把握できる。

【0047】

(実施例 3)

本実施例は、実施例 2 において、同一仮想計算機システム内に設けた特定の L P A R 上で動作するプログラムに再構成方針生成部 206 と負荷状態監視部 207 を実装したシステムの例である。

【0048】

図 19 に本実施例の構成を示す。L P A R 5 x 上で実行する監視プログラム 190 が、L P A R 50、・・・、L P A R 5 k の負荷状態の監視と物理資源の再構成要求の発行を行う。

【 0 0 4 9 】

L P A R 5 x 上の監視プログラム 1 9 0 は、負荷状態調査要求を各 L A P R 5 0、・・・、L P A R 5 k に対して転送する。このとき、L P A R 間の通信は、特開平 1 0 - 3 0 1 7 9 5 に示されているように、ハイパーバイザで仮想的に通信をエミュレートする方法、I O チャンネルを使用する方法、L P A R 内は C P U で L P A R 外の計算機との通信はチャンネルを使用する方法が知られている。本実施例では、L P A R 間通信に際しては如何なる方法をとってもよく、本発明では詳細については省略する。ここでは、ハイパーバイザが L P A R 間の通信路をエミュレートする例を使用するとする

(負荷状態の取得)

監視プログラム 1 9 0 は他の L P A R 5 0、・・・、L P A R 5 k の負荷状態を要求する(5 0 0)。要求を受信した各 L P A R は、負荷情報(実施例 1 の C P U 使用率、実行待ち行列長、実施例 2 のアプリケーションの処理の応答時間)を L P A R 5 x に転送する(5 0 1)。この負荷状態調査要求 5 0 0 の発行タイミング 5 1 0 は、オペレータによって監視プログラム 1 9 0 内に設定されている。

【 0 0 5 0 】

(再構成要求の発行)

監視プログラム 1 9 0 には、実施例 1 の負荷状態監視部 2 0 7 と同様にオペレータによってあらかじめ負荷の閾値 5 1 1 が設定されプログラム内部に保持している。この閾値 5 1 1 を超える負荷が監視された場合、監視プログラム 1 9 0 はハイパーバイザ 4 0 に対して現在の資源の割り当てを通知する要求を発行し(5 0 2)、資源割り当て情報をハイパーバイザ 4 0 から受け取る(5 0 3)。負荷状態と C P U の割り当て時間や C P U 数などの構成を変更する方針の組み合わせを記した負荷対策表 5 1 2 は、オペレータによって監視プログラム 1 9 0 に設定されている。この負荷対策表 5 1 2 と負荷状態から新たな資源割り当ての方針を示す再構成方針テーブル 5 1 3 を生成する。再構成方針テーブル 5 1 3 は、実施例 1 で示した方法で生成されるため、本実施例では説明を省略する。

【 0 0 5 1 】

次に、監視プログラム 1 9 0 は、ハイパーバイザ 4 0 に対し再構成方針テーブ

ル 5 1 3 と、L P A R 5 0、・・・、5 k に割り当てている資源の再構成を要求するコマンドを含む再構成要求 5 0 5 を発行する。ハイパーバイザ 4 0 は、再構成完了後、再構成完了通知 5 0 6 を監視プログラム 1 9 0 へ転送する。以上で、負荷状態を反映した L P A R の再構成が完了する。そして、監視プログラム 1 9 0 は各 L P A R の負荷状態の監視を再開する。

【 0 0 5 2 】

以上のように、同一仮想計算機システム内に設けた特定の L P A R 上で動作する監視プログラムに再構成方針生成部 2 0 6 と負荷状態監視部 2 0 7 を実装したシステムを示した。これによると、再構成方針のアルゴリズムをオペレータが自由に変更できるように設定した場合でも、上記アルゴリズムはハイパーバイザにはないため、システムの根幹であるハイパーバイザをオペレータが操作できるようにする必要がない。従って、セキュリティの問題や、オペレータがハイパーバイザに異常をもたらすような操作を行なう惧れがない。また、再構成方針に異常が起こってもハイパーバイザで不当な処理を監視するようにすれば、システム全体の異常が引き起こされることがない。

【 0 0 5 3 】

(実施例 4)

本実施例は、実施例 3 において、同一仮想計算機システム内に設けた特定の L P A R 上で動作する監視プログラム 1 9 0 を、別の物理計算機上に実装したシステムの例である。

【 0 0 5 4 】

図 2 0 に本実施例の構成を示す。物理計算機 6 0 - x の L P A R 6 0 x - x 上で実行する監視プログラム 1 9 0 が、物理計算機 6 0 - 0 の L P A R 6 0 0 - 0、・・・、L P A R 6 0 0 - k の負荷状態の監視と物理資源の再構成要求の発行を行う。

【 0 0 5 5 】

L P A R 6 0 x - x 上の監視プログラム 1 9 0 は、負荷状態調査要求 5 0 0 - A を物理計算機 6 0 - 0 に対して発行する。物理計算機 6 0 - 0 のハイパーバイザ 4 0 - 0 は各 L A P R 6 0 0 - 0、・・・、L P A R 6 0 0 - k に対してこの

負荷状態調査要求を転送する。物理計算機 6 0 - 0 (6 0 - x) 上のハイパーバイザ 4 0 - 0 (4 0 - x) は I O チャンネルを使用して別の物理計算機 6 0 - x (6 0 - x) と通信をする。本実施例では監視プログラム 1 9 0 は L P A R 上のプログラムとして実装されているが、物理計算機 6 0 - x 上には仮想計算機システムではなく単一の計算機が動作していてもよい。つまり、L P A R 6 0 x - x が単一の物理計算機となってもよいということである。

【 0 0 5 6 】

(負荷状態の取得)

監視プログラム 1 9 0 は他の物理計算機 6 0 - 0 上に実装された L P A R 6 0 0 - 0 、 . . . 、 L P A R 6 0 0 - k の負荷状態を I / O 5 2 0 - x を通して要求する (5 0 0 - A 、 5 0 1 - A) 。要求を受信した各 L P A R は、負荷情報 (実施例 1 の C P U 使用率、実行待ち行列長、実施例 2 のアプリケーションの処理の応答時間) を L P A R 6 0 x - x に転送する (5 0 0 - B 、 5 0 1 - B) 。この負荷状態調査要求 5 0 0 - A 、 5 0 0 - B の発行タイミング 5 1 0 は、オペレータによって監視プログラム 1 9 0 内に設定されている。

【 0 0 5 7 】

(再構成要求の発行)

監視プログラム 1 9 0 には、実施例 1 の負荷状態監視部 2 0 7 と同様にオペレータによってあらかじめ負荷の閾値 5 1 1 が設定されプログラム内部に保持している。この閾値 5 1 1 を超える負荷が監視された場合、監視プログラム 1 9 0 はハイパーバイザ 4 0 - 0 に対して現在の資源の割り当てを通知する要求を発行し (5 0 2 - A) 、資源割り当て情報をハイパーバイザ 4 0 - 0 から受け取る (5 0 2 - B 、 5 0 3 - B) 。負荷状態と C P U の割り当て時間や C P U 数などの構成を変更する方針の組み合わせを記した負荷対策表 5 1 2 は、オペレータによって監視プログラム 1 9 0 に設定されている。この負荷対策表 5 1 2 と負荷状態から新たな資源割り当ての方針を示す再構成方針テーブル 5 1 3 を生成する。再構成方針テーブル 5 1 3 は、実施例 1 で示した方法で生成されるため、本実施例では説明を省略する。

【 0 0 5 8 】

次に、監視プログラム 1 9 0 は、ハイパーバイザ 4 0 - 0 に対し再構成方針テーブル 5 1 3 と物理計算機 6 0 - 0 上の L P A R 6 0 0 - 0、・・・、6 0 0 - k に割り当てている資源の再構成を要求するコマンドを含む再構成要求 5 0 4 - A、5 0 4 - B を発行する。ハイパーバイザ 4 0 - 0 は、再構成完了後、再構成完了通知 5 0 5 - A、5 0 5 - B を監視プログラム 1 9 0 へ転送する。以上で、負荷状態を反映した L P A R の再構成が完了する。そして、監視プログラム 1 9 0 は各 L P A R の負荷状態の監視を再開する。

【 0 0 5 9 】

以上のように、監視プログラムを別の物理計算機に実装した例を示した。これによれば、L P A R が実装された他の物理計算機の管理も集中して行なうことが可能になるという効果がある。

【 0 0 6 0 】

(実施例 5)

実施例 2 では、アプリケーションプログラムの処理の応答時間を調査するために、ハイパーバイザから O S への割り込みを起こし、O S はアプリケーションプログラムにシグナルを送ってアプリケーションプログラムで計測している処理の応答時間を要求した。ところが、本実施例ではアプリケーションプログラムに対し特殊なインタフェースを持たずに L P A R 上で動作するアプリケーションプログラムの負荷状況を調査する方法を図 2 1 を用いて説明する。ここでは、アプリケーションプログラムは応答時間を計測していない、又は計測していてもそれを読み出すインタフェースがない場合である。アプリケーションの負荷状態を調査した後、L P A R の物理資源割り当て変更手順は実施例 4 に従うので、本実施例で L P A R 再構成の説明は省略する。

【 0 0 6 1 】

図 2 1 は、物理計算機 6 0 - 0、6 0 - x とそれらを結合するネットワーク 6 1 で構成され、各物理計算機上では L P A R 6 0 0 - 0、6 0 0 - k、6 0 x - 0、6 0 x - k、6 0 x - x が稼動している。物理計算機 6 0 - 0 上の L P A R 6 0 0 - 0 では、WWW(W o r l d W i d e W e b)サーバといったアプリケーションプログラム 1 9 5 が動作している。物理計算機 6 0 - x 上の L P A R 6

0x-xで動作している監視プログラム190がアプリケーションプログラム195にデータのアクセス要求700を発行する。アプリケーションプログラム195がWWWサーバであれば、ホームページの読み出し要求を発行する。アプリケーションプログラム195は要求700に対して応答701を発行する。監視プログラム190では要求700発行から応答703受信までの応答時間を応答時間履歴703に記録する。要求700は、アプリケーションプログラム195の性能を低下させない程度の間隔で発行するかあるいは監視プログラム内にオペレータがあらかじめ設定しておくようにする(図示せず)。

【0062】

監視プログラム190は、応答時間履歴703の推移を観測し、応答時間が大きい状態が継続したときハイパーバイザ40-0に対し、応答時間が大きくなっているアプリケーションプログラムが動作しているLPARの物理資源の割り当てを多くするように要求する。この資源割り当て変更の手順は実施例4に従う。また、監視プログラム190で応答時間履歴を比較的長い時間(数日間)採取し、負荷の変動の規則性を探し、周期に合わせた形でLPARの物理資源配分を計画的に変更させてもよい。

【0063】

以上のように、アプリケーションプログラムが応答時間を計測していない、又は計測結果を読み出すインターフェイスがない場合、監視プログラムがデータのアクセス要求を出して、その応答を受け取るまでの時間を計る。そして、応答時間の履歴を格納しておくことによって、アプリケーションプログラムの応答時間を把握する例を示した。これによって、アプリケーションプログラムが応答時間を計測していない場合やインターフェイスがない場合でも応答時間の把握が出来る。

【0064】

(実施例6)

実施例2、3では、負荷状況に対して対策案をあらかじめ決めていたが、物理資源の割り当てで可能な対策とその優先順位を列挙した表(図22)を作成し、図23の手順で順次対策を行う方法もある。以降図23で示した手順を説明する。

【0065】

LAPRの負荷状況をこれまでの実施例で示した方法で採取し、オペレータがあらかじめ決めていた負荷状態の閾値を超えた場合、LAPRの再構成の準備を開始する(800)。いま、対策案は全部で N_{max} 個あるとする。まず図22の優先順位1(801)の対策を実施する(802)。対策後のLAPRで運用した後、負荷が改善しなかった場合は、再構成前の構成に戻す(806)。全ての対策案が完了しているかどうか確認(804)し、すべて完了していない場合、次の対策案(805)を実施する。全ての案が終了した後、効果があった対策が存在したか確認(806)し、存在しなかった場合はオペレータに対して負荷対策が不能であったことを、画面表示やログファイル、あるいはブザーで通知する(809)。このフローに従えば、効果がある限り複数の対策が合わせて実行される。

【0066】

以上のように、複数の対策案が優先順位と共に用意され、負荷の低減に寄与する対策が1又は複数採用される手法を示した。これによれば、負荷低減に効果のある対策が試行によって選択される。

【0067】

(実施例7)

また、昼と夜で負荷が大きく変化するような利用形態を考えたとき、負荷が高い時間帯は他のLAPRから資源を集め、夜は他のLAPRに資源の一部を開放するというように計画的に構成を変化する方法がある。このような、負荷変位の規則性を見つける手段と実施例1を組み合わせる方法もある。以降では、負荷の規則的変化を発見する手段と実施例1を組み合わせた実施例を示す。

【0068】

負荷の規則的変化を発見する方法の1つとして、数日間の負荷を記録し、同一時間帯で負荷の高低を負荷状態監視部で調査する。例えば、図12に示すように数日間の同一時刻の負荷変動の平均値の変化を調査する。そして、負荷の閾値を設定し、高い負荷の時間帯では物理資源の割り当てを多くし、負荷が低い負荷の時間帯では、他のLAPRに資源を提供する。また、それ以外の時間帯では初期に設定した資源配分に戻すというようなスケジューリングを負荷状態監視部20

7で行う。資源の配分方法は、高負荷の閾値(実行待ち行列長だと例えば3)以上になっている時間帯での最大負荷となったときのシステムの状態を基準にして再構成方針生成部206で実施例1で示した方法で再構成案を生成し、資源の動的再構成を行う。さらに、このような周期的に資源配分を変更するだけでなく、負荷状態に対して動的に資源分配を行い負荷軽減の微調整を行ってもよい。

【0069】

また、負荷の周期性を求める手段として、解析的に負荷の規則的变化を発見する方法を用いてもよい。ここでは一例としてFFT(高速フーリエ変換)を利用し、負荷の変化の周期を解析的にもとめる方法を示す。FFTのアルゴリズムについては「デジタル信号処理の基礎 辻井重男監修 電子情報通信学会」(1988. 3. 15初版発行)などの信号処理の教科書で扱われている有名なアルゴリズムなので具体的な説明は省略する。図13のようなT時間で32個の測定点をもつ時系列の負荷状態が与えられたとする。ただし、ここではプロセスの実行待ち行列長を負荷の例に取り上げている。図13をFFTでスペクトル分布(信号処理では電力スペクトル分布に相当、ここでは単にスペクトルと表現している)を計算すると図14のようになる。標本化定理により分析できる高調波の次数は16である。負荷の規則変化を調査しているので直流成分に相当する周波数0の部分は無視をすると、最もスペクトルの強度が大きいのは次数3の周波数($= 3/2\pi T$)である。そこで、 $3/2\pi T$ の周波数で負荷が変動していると仮定し、L P A Rへの物理資源割り当てを変化させる。このとき、図13で最大の負荷時の数値を元に物理資源分配構成を作成しておく。そして、最小負荷と最大負荷の中間値を閾値とし、負荷が増大し、閾値に達したときに、先に生成しておいた構成にL P A Rを再構成する。本実施例では半周期ずつ構成を変更する場合を示したが、さらに細かく分類して物理資源のL P A Rへの分配してもよい。L P A Rの再構成手順は実施例1に示したとおりである。

【0070】

以上のように負荷の周期性を求める手段として、解析的に負荷の規則的な変化を発見する方法、具体的にはFFT、を用いたことにより、負荷の変動をオペレータなど人の主観に頼らずに正確に求めることが出来る。

【 0 0 7 1 】

(実施例 8)

実施例 1 から 7 では、CPU (物理プロセッサ) に関する負荷を例に挙げてきた。主記憶に関する負荷状態で LPAR の資源割り当てを変化させてもよい。また、CPU と主記憶の資源割り当てを、同時に対策しても構わない。

【 0 0 7 2 】

主記憶の負荷状態を表す指標に、スワップやページングの回数がある。これらに関しても実施例 1 の CPU 負荷の場合と同様に負荷状態の監視をし、負荷状態が高い LPAR に割り振る主記憶を増やすように LPAR を動的再構成をする。主記憶の量を変化させる LPAR は、実施例 1 の CPU 数を増やす場合と同様に一旦 LPAR 上で稼動する OS をストップ状態にし、物理資源を分配 (ここでは主記憶量の変更) をした後、LPAR 制御部 202 が、論理プロセッサ制御部 203 を介してもしくは直接、新たに割り当てられた主記憶を割り込み等で LPAR 上の OS に通知し、LPAR 上の OS が自発的に新しく追加された主記憶の拡張をするコマンドを、対応する LPAR 制御部 202 に送る。本例でも実施例 2、実施例 3 の構成変更方法を適用してもよい。

【 0 0 7 3 】

以上のように、割り当てられた主記憶の容量が不足しているかどうかの判断が出来る。

【 0 0 7 4 】

(実施例 9)

本実施例では、データセンタにおいて仮想計算機システムを使用する例を示す。データセンタ運営者は、図 24 に示す契約表の内容で各顧客と契約を結ぶ。契約等級 1000 は A、B、C の順番で優先順位が決まり、A の契約が最も優先順位が高い。契約等級 1000 ごとに、契約料 1001 が PA、PB、PC と定められる。顧客とは、契約の優先順位が高いほどアプリケーションの応答時間などの性能保証を優先して行う契約を結ぶ。

【 0 0 7 5 】

データセンタ運営者は、図 26 に示す様に各顧客 1005 ごとに LPAR の割

り当て 1 0 0 7 と契約等級 1 0 0 6 を設定する。各顧客のアプリケーションプログラムはこの L P A R 上で動作する。優先度 1 0 0 2 が高い L A P R ほど、優先して物理資源を割り当てていく。データセンタ運営者は、図 2 5 の表に従い各契約等級 1 0 0 0 ごとに優先順位 1 0 0 2、L P A R の負荷状態の上限閾値 1 0 0 3、下限閾値 1 0 0 4 を決定する。上限閾値 1 0 0 3 とは L P A R 上で動作する O S やアプリケーションの負荷が大きくなったとき許容可能な最大値であり、L A P R への資源割り当ての増大を要求する契機を判断するための数値である。下限値 1 0 0 4 は、負荷がこの数値より小さいときは、L P A R に割り当てられた資源量を初期値に戻すための契機を判断するための数値である。図 2 5 の表は、仮想計算機の負荷状態を監視する手段の内部にも記憶されている。

【 0 0 7 6 】

図 2 7 を使って、データセンタの運営の流れを説明する。まず、実施例 1 から 8 で示してきた L P A R の負荷状態を監視する手段において、L P A R の負荷状態を観測する (9 5 0)。負荷状態で図 2 5 の上限閾値 1 0 0 3 を超えた L P A R があるかどうか確かめる (9 5 1)。上限閾値 1 0 0 3 を超えた L P A R がなければ L P A R の構成変更をすることなく運用を継続するが、上限を超えた L P A R があり、かつ全ての L P A R で負荷が高くなっていないければ (9 5 2)、負荷への対策の余地があると考えられる。

【 0 0 7 7 】

高負荷な L P A R への資源割り当てを開始する前に、負荷が下限閾値を超えていない L P A R に割り当てられている物理資源の解放を行う (9 5 3)。このとき、L P A R へ割り当てられている資源の量は初期設定の数値に戻す。次に、上限閾値を超えた L P A R で最も優先順位が高いものに、先ほど開放した物理資源や、上限閾値を超えてない優先順位の低い L P A R から一部資源を移す (9 5 4)。この資源を移すアルゴリズムは実施例 1 の方法を使ってもよい。このようにして L A P R の資源割り当てを変更していく。

【 0 0 7 8 】

以上のように、契約等級によって優先的に資源を割り当てる例を示した。これによれば顧客の特性と契約料金に応じたサービスができる。

【 0 0 7 9 】

【発明の効果】

本発明によれば、仮想計算機システムの負荷状況をより正確に把握し、各 L P A R への物理資源配分を自動的に決定して L P A R の再構成を実行する、管理が容易な仮想計算機システムを提供出来た。

【図面の簡単な説明】

【図 1】

本発明の一実施例を説明する図。

【図 2】

本発明の実施例における 1 つの仮想計算機システムを構成する物理計算機システムの構成例を示す図。

【図 3】

本発明の実施例における仮想計算機システムの概要を示す図。

【図 4】

本発明の実施例における主記憶装置の領域の配分を示す図。

【図 5】

本発明の実施例における L P A R 情報テーブルを示す図。

【図 6】

本発明の実施例におけるハイパーバイザの構成を示す図。

【図 7】

本発明の実施例における L P A R 毎の C P U の割り当て率を規定するテーブルを示す図。

【図 8】

本発明の実施例における L P A R 毎の C P U の負荷状況を表わすテーブルを示す図。

【図 9】

本発明の実施例における再構成方針テーブルを示す図。

【図 1 0】

本発明の実施例における負荷対策表を示す図。

【図 1 1】

本発明の実施例における CPU 割り当て時間引当表を示す図。

【図 1 2】

本発明の実施例におけるある LPAR の平均 CPU 負荷の変動を示す図。

【図 1 3】

本発明の実施例におけるある LPAR の CPU 負荷のサンプリングデータを示す図。

【図 1 4】

本発明の実施例における CPU 負荷のサンプリングデータのスペクトル分布を示す図。

【図 1 5】

本発明の実施例におけるポリシーサーバの実装例を示す図。

【図 1 6】

ポリシーサーバの他の実装例を説明するための図。

【図 1 7】

本発明の他の実施例を説明する図。

【図 1 8】

本発明の実施例における LPAR 毎のアプリケーション平均応答時間テーブルを示す図。

【図 1 9】

本発明の実施例における再構成方針生成部と負荷状態監視部の他の実装例を示す図。

【図 2 0】

本発明の実施例における再構成方針生成部と負荷状態監視部の更に他の実装例を示す図。

【図 2 1】

本発明の実施例におけるアプリケーションプログラムの応答時間の他の取得方法を説明する図。

【図 2 2】

本発明の実施例における負荷状況に対する対策内容を表わした表を示す図。

【図 2 3】

図 2 2 に従って順次対策を行なう処理を示すフローチャート。

【図 2 4】

データセンタにおけるユーザとの契約等級と契約料の対応を表わす表を示す図。

【図 2 5】

データセンタにおける契約等級と優先度と負荷の上限と下限の閾値との対応を表す表を示す図。

【図 2 6】

データセンタにおける顧客と契約等級と使用 L P A R との対応を表わす表を示す図。

【図 2 7】

データセンタにおける管理プログラムのフローチャート。

【符号の説明】

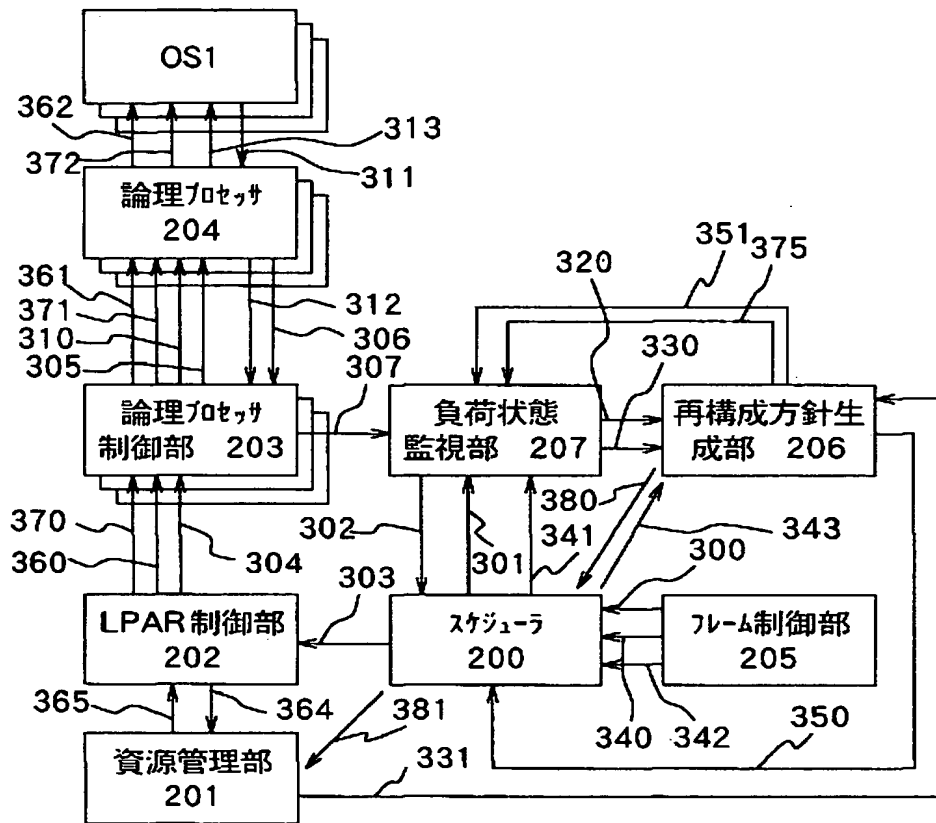
- 1 : オペレーティングシステム(O S)、
- 1 0、1 1、・・・、1 n : C P U 0、C P U 1、・・・、C P U n、
- 2 0 主記憶装置、
- 3 0、3 1、・・・、3 m : I / O 装置 I / O 0、I / O 1、・・・、I / O m、
- 4 0、4 0 - 0、4 0 - x : ハイパーバイザ、
- 5 0、・・・、5 k : 仮想計算機 L P A R 0、・・・、L P A R k、
- 5 0 - 0、・・・、5 0 - n : L P A R 0 下の論理プロセッサ L P 0、・・・、L P n、
- 5 k - 0、・・・、5 k - n : L P A R k 下の論理プロセッサ L P 0、・・・、L P n、
- 6 0 - 0、6 0 - x : 物理計算機、
- 6 1 : ネットワーク、
- 1 0 0 : L P A R 情報テーブル、

1 9 0 : 監視プログラム、
2 0 0 : スケジューラ、
2 0 1 : 資源管理部、
2 0 2 : L P A R 制御部、
2 0 3 : 論理プロセッサ制御部、
2 0 4 : 論理プロセッサ (L P)、
2 0 5 : フレーム制御部、
2 0 6 : 再構成方針制御部、
2 0 7 : 負荷状態監視部、
4 0 0 : アプリケーション、
6 0 0 - 0、6 0 0 - k、6 0 x - 0、6 0 x - k、6 0 x - x : L P A R、

【書類名】 図面

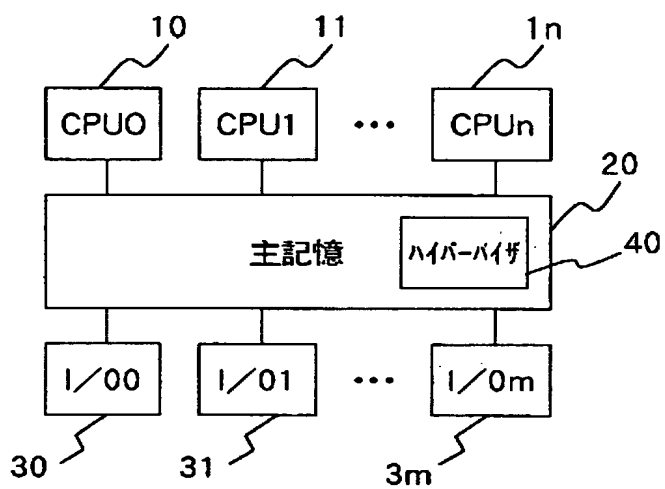
【図 1】

図 1



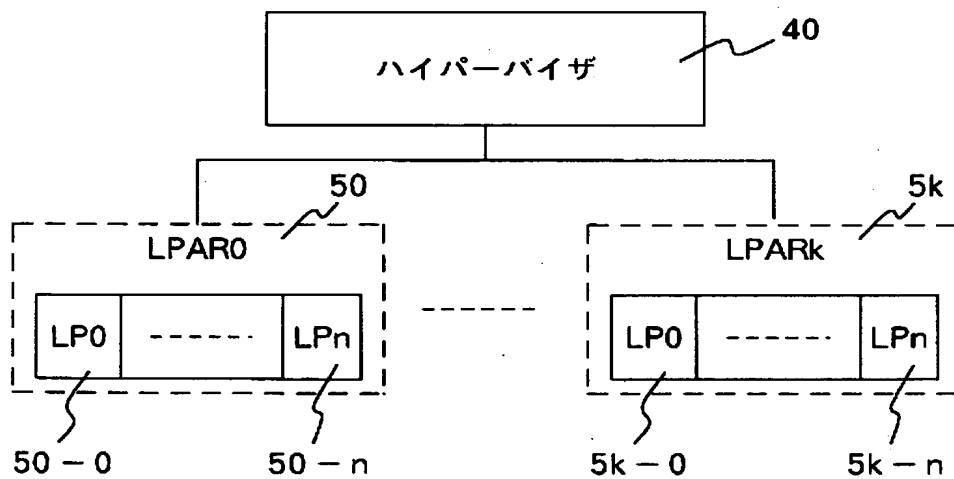
【図 2】

図 2



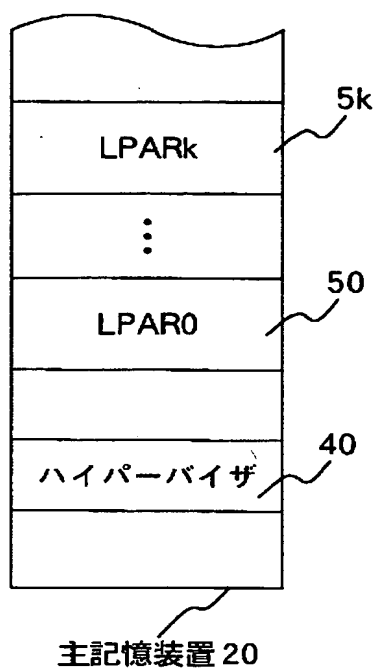
【図 3】

図 3



【図 4】

図 4



【図 5】

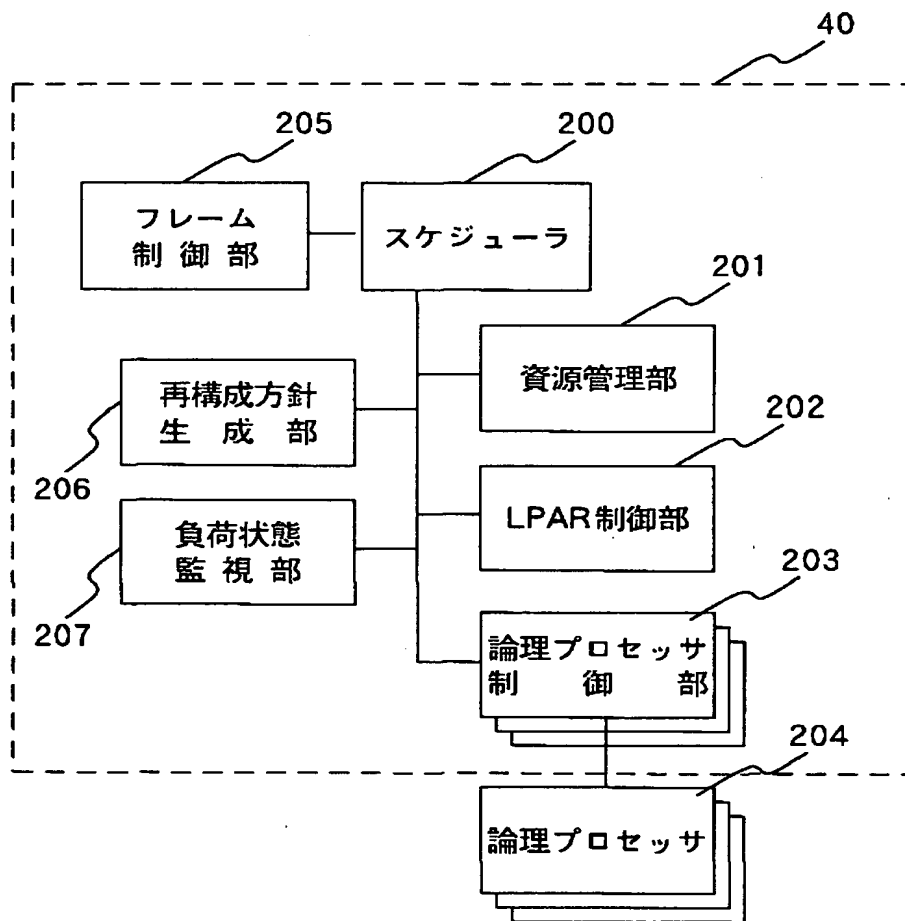
図 5

LPAR名	開始 アドレス	主記憶 容 量	CPU0	CPU1	...	CPU _n
LPAR ₀	a	m ₁	100	50	...	—
⋮						
LPAR _k	i	m _k	—	50	...	100

100
LPAR情報テーブル

【図 6】

図 6



【図 7】

図 7

CPU 割り当て (単位%)

	CPU0	CPU1	CPU2	CPU3
LPAR0	100	50		
LPAR1		50	100	
LPAR2				100

【図 8】

図 8

CPU 負荷状況

	CPU 使用率 [単位%] 実行待ち行列長			
	CPU0	CPU1	CPU2	CPU3
LPAR0	100/5	100/4	-	-
LPAR1	-	10/0	5/0	-
LPAR2	-	-	-	20/0

【図 9】

図 9

新規 CPU 割り当て案 (単位%)

	CPU0	CPU1	CPU2	CPU3
LPAR0	100	75	50	30
LPAR1		25	50	
LPAR2				70

再構成方針テーブル 900

【図 1 0】

図 10

負荷対策表

現 象	優先順位	対 策
CPU 使用率が閾値以上 かつ実行待ち行列長 閾値以上	1	CPU 割り当て時間増加、かつ自 LPAR で未使用の CPU 追加
	2	自 LPAR で未使用の CPU 追加
	3	CPU 割り当て時間増加
	⋮	
CPU 使用率閾値以上 かつ実行待ち行列長 閾値以下	1	CPU 割り当て時間増加
	2	自 LPAR で未使用の CPU 追加
	⋮	
⋮	⋮	

【図 1 1】

図 11

CPU 割り当て時間引当表

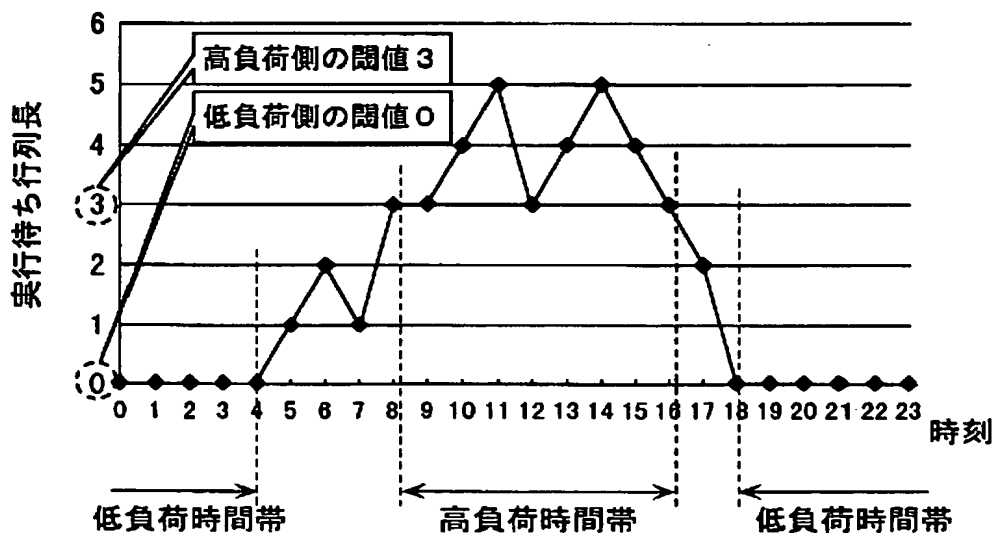
CPU 時間を渡す LPAR の CPU 使用率 (%)	他 LPAR へ割り当てる CPU 割り当て時間の割合 (%)
0 以上 10 未満	50
10 以上 20 未満	40
20 以上 30 未満	30
30 以上 40 未満	20
40 以上	0

(引き渡す CPU 割り当て時間) = (現在設定されている CPU 割り当て時間)
× (他 LPAR へ割り当てる CPU 割り当て時間)

【図 1 2】

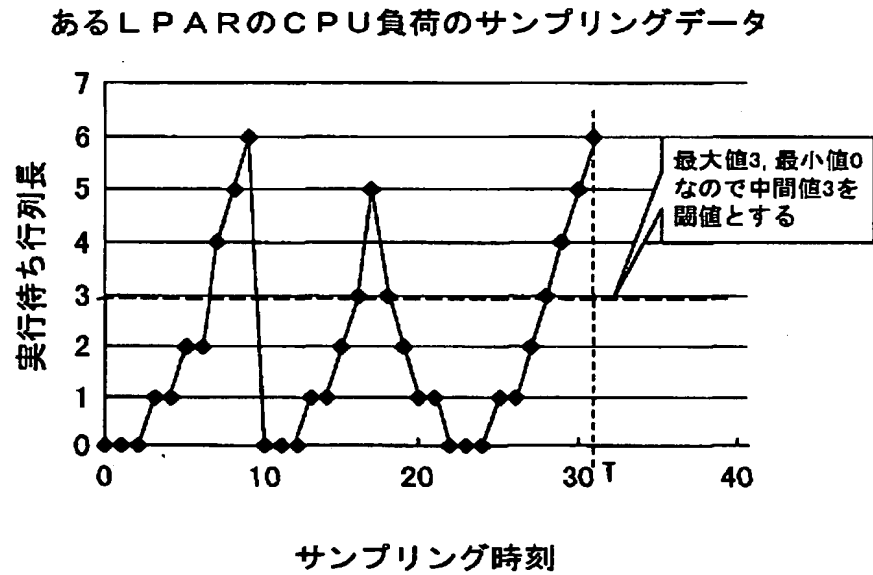
図 12

ある LPAR の数日間の平均 CPU 負荷の変動
(負荷として実行待ち行列長を使用した例)



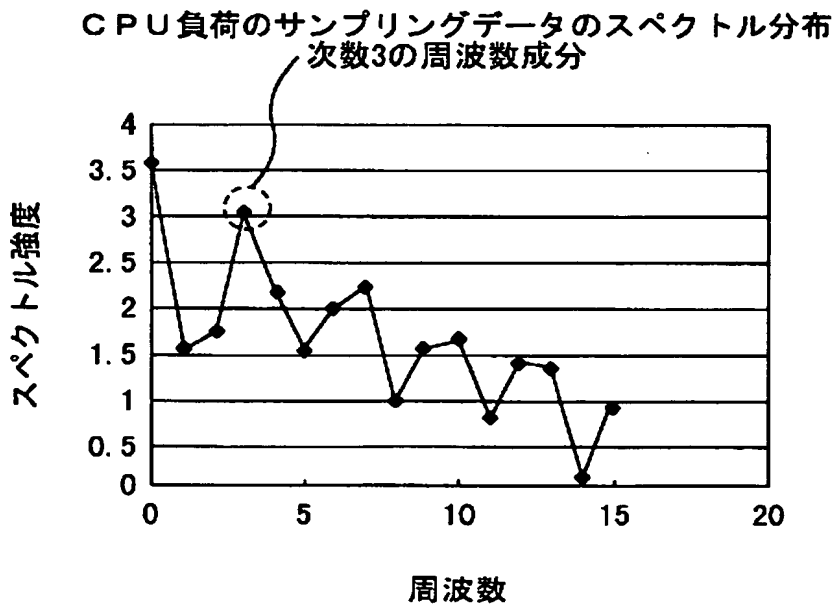
【図 1 3】

図 13



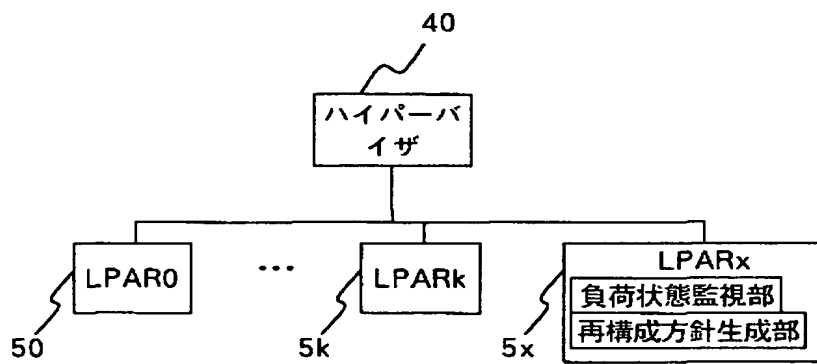
【図 1 4】

図 14



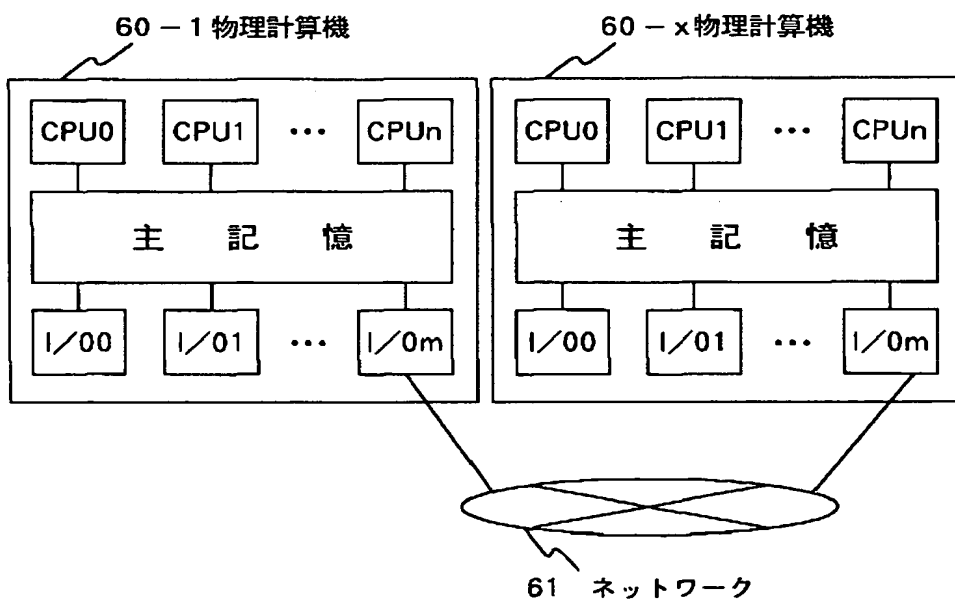
【図 15】

図 15



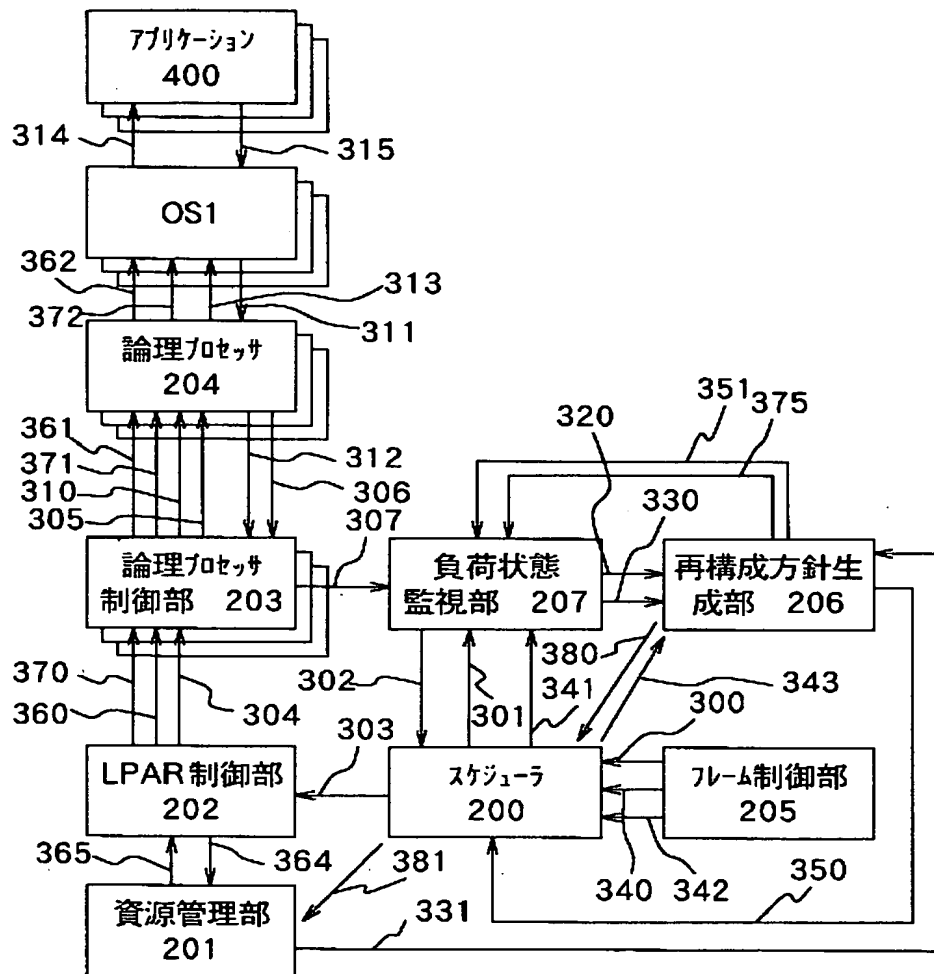
【図 16】

図 16



【图 1 7】

圖 17



【図 1 8】

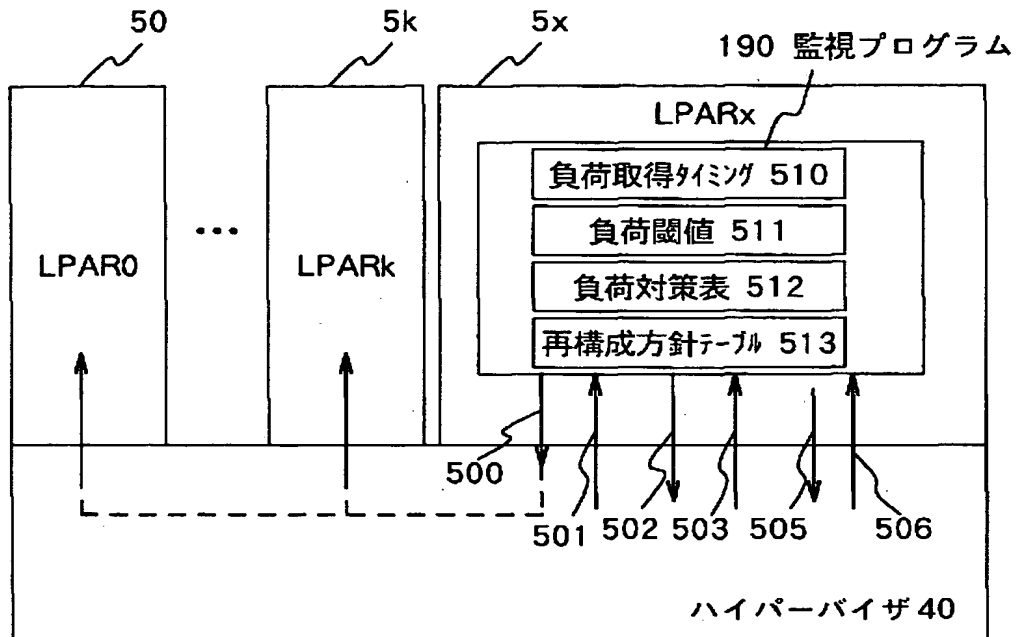
図 18

アプリケーション平均応答時間

	平均応答時間[秒]
LPAR0	10
LPAR1	2
LPAR2	3

【図 1 9】

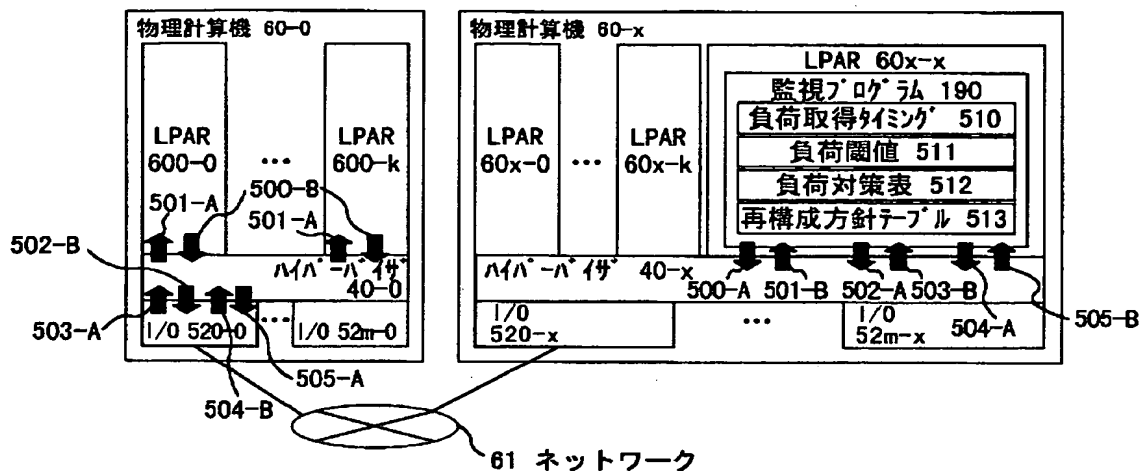
図 19



- 500 負荷状態調査要求
- 501 負荷状態情報
- 502 資源割り当て読み出し要求
- 503 資源割り当て情報
- 505 再構成要求
- 506 再構成完了通知

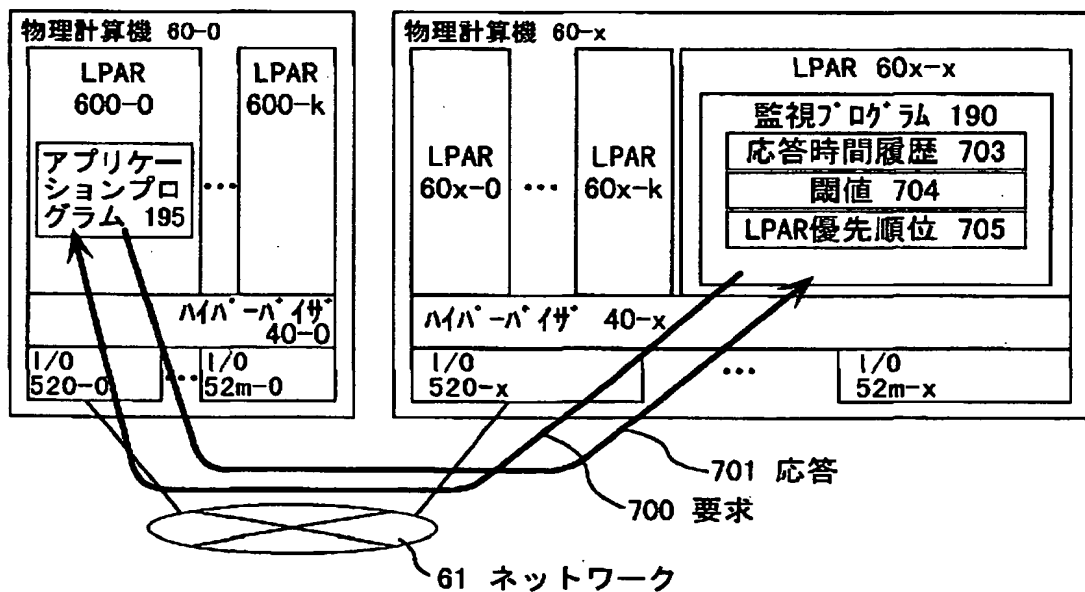
【図 2 0】

図 20



【図 2 1】

図 21



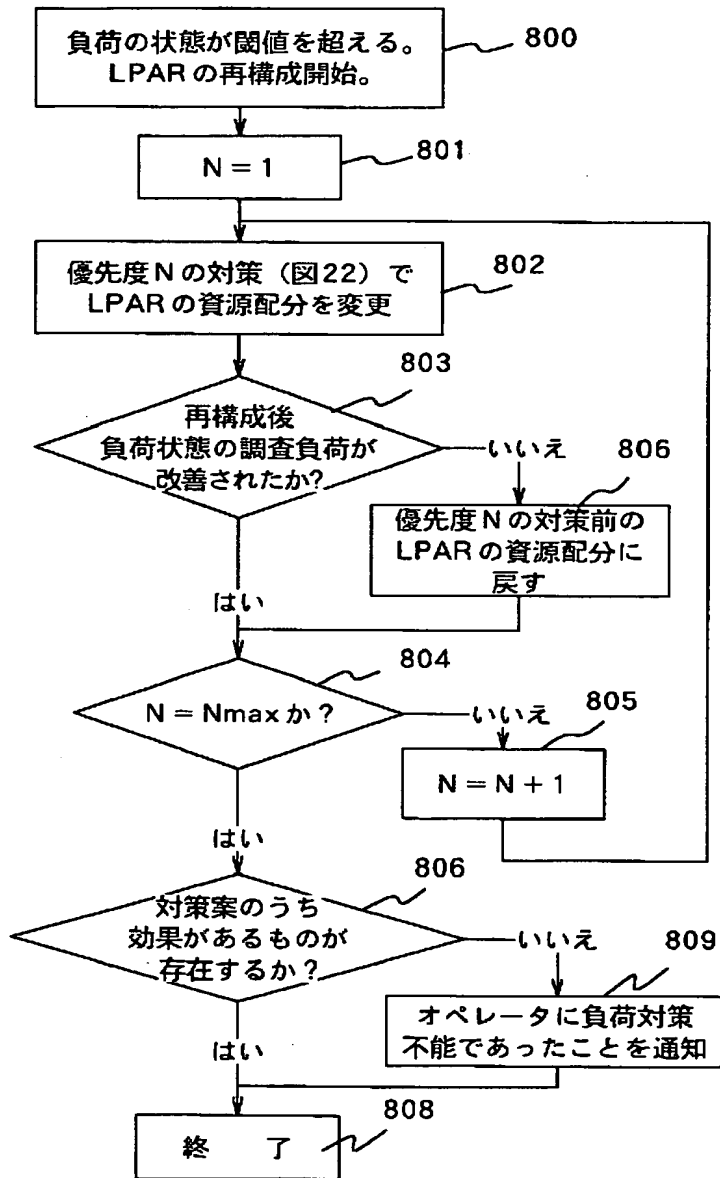
【図 2 2】

図 22

優先順位	対策内容
1	CPU 割り当て時間の増加
2	CPU 数の増加
3	主記憶の増加
4	ディスクのスワップ領域拡大
⋮	⋮

【図 23】

図 23



【図 2 4】

図 24

契約等級 1000	契約料 1001
A	PA
B	PB
C	PC

【図 2 5】

図 25

契約等級 1000	契約料 1002	上限閾値 1003	下限閾値 1004
A	1	UTA	LTA
B	2	UTB	LTB
C	3	UTC	LTC

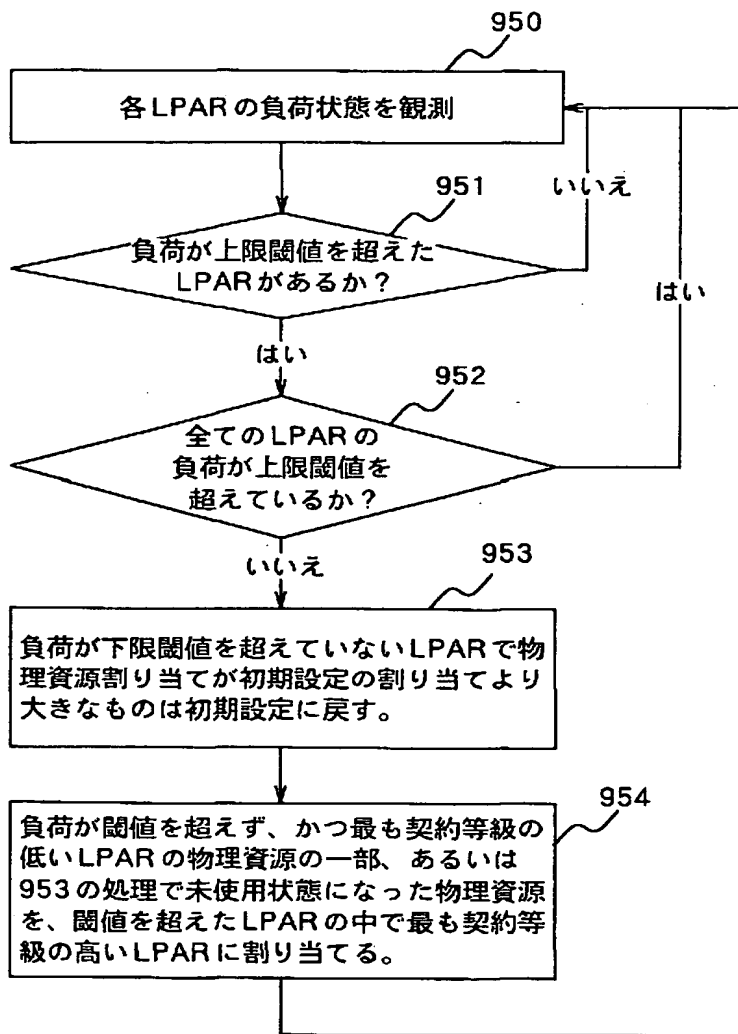
【図 2 6】

図 26

顧客 1005	顧客契約等級 1006	使用 LPAR 1007
C0	B	LPAR0
C1	A	LPAR1
C2	C	LPAR2
C3	C	LPAR3

【図 27】

図 27



【書類名】 要約書

【要約】

・【課題】仮想計算機システムの負荷を反映して、各 L P A R への物理資源配分を自動的に決定し、L P A R の再構成を実行することは困難であった。

【解決手段】物理計算機を構成する物理資源を、排他的にまたは時分割的に論理的に分割することにより複数の L P A R が動作し、各 L P A R 間で物理資源の割り当てを動的に変更する再構成手段とを持つ仮想計算機システムにおいて、各 L P A R のアプリケーションや O S で計測された負荷状況に基づき、各 L P A R への物理資源の配分を決定し、L P A R の再構成を行う。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日	1990年 8月31日
[変更理由]	新規登録
住 所	東京都千代田区神田駿河台4丁目6番地
氏 名	株式会社日立製作所